Knowledge Acquisition for Temporal Abstraction

Adam Stein, Mark A. Musen, M.D., Ph.D., Yuval Shahar, M.D., Ph.D.
Section on Medical Informatics, Stanford University School of Medicine,
Stanford, CA 94305-5479

Temporal abstraction is the task of detecting relevant patterns in data over time. The knowledge-based temporal-abstraction method uses knowledge about a clinical domain's contexts, external events, and parameters to create meaningful interval-based abstractions from raw time-stamped clinical data. In this paper, we describe the acquisition and maintenance of domain-specific temporal-abstraction knowledge. Using the PROTÉGÉ-II framework, we have designed a graphical tool for acquiring temporal knowledge directly from expert physicians, maintaining the knowledge in a sharable form, and converting the knowledge into a suitable format for use by an appropriate problem-solving method. In initial tests, the tool offered significant gains in our ability to rapidly acquire temporal knowledge and to use that knowledge to perform automated temporal reasoning.

ACQUISITION OF TEMPORAL KNOWLEDGE

The temporal-abstraction task, which, in a clinical setting, consists of creating interval-based abstractions from time-stamped interventions and clinical parameters (Figure 1), is highly relevant to any domain in which patient data are tracked over time. The temporal-abstraction task can be solved independently of the particular clinical domain by a method called **knowledge-based temporal abstraction (KBTA).**[1] The KBTA method uses an extensive domain model, or **ontology,**[2] to represent formally the terms, concepts, and relations relevant to the temporal-abstraction task. This explicit declaration of knowledge requirements offers several potential benefits, such as increased reuse of the KBTA method across multiple clinical domains, easier acquisition and maintenance of domain-specific knowledge bases, and increased sharing of those knowledge bases with other tasks in the same domains. However, acquisition and maintenance of the domain-specific knowledge necessary for the operation of the KBTA method remain significant difficulties. In particular, we must address two problems: (1) we must design a KA tool that allows expert physicians to enter, browse, and update the clinical knowledge in the system directly and easily; and (2) we must convert, or map, the acquired knowledge into a form readable by the method implementation.

Mapping knowledge bases to problem-solving methods is an important step in facilitating reuse.[3] Often, method implementations are legacy software that cannot or should not be modified to accommodate each new change in a knowledge base. Moreover, a single method implementation is ideally applicable to knowledge bases from different domains, in which case mapping is essential. Note that **RÉSUMÉ**, our implementation of the KBTA method, is independent of both domain (e.g., oncology) and application (e.g., guideline-based therapy). The KA tool generated for a particular application, however, is typically tailored to that application. Thus, our intent is to leave intact the implementations of domain-independent methods such as RÉSUMÉ, while tailoring KA tools to certain users and applications. We must then resolve the differences between a method's ontology and a KA tool's ontology.

We have addressed the problem of knowledge acquisition and maintenance by generating a KA tool within the PROTÉGÉ-II[4] development environment. In addition, we have designed a flexible and reusable filter that performs the necessary mapping between the KA tool's output and RÉSUMÉ's internal knowledge structures. The tool enables us to acquire temporal-abstraction knowledge rapidly and to apply RÉSUMÉ to a variety of tasks and clinical domains.

KNOWLEDGE-BASED TEMPORAL ABSTRACTION

To provide a background for understanding the challenges and requirements of designing a KA tool for temporal abstraction, we shall describe briefly the relevant aspects of the KBTA method. The KBTA method decomposes the temporal-abstraction task into five parallel **subtasks** (Figure 2): 1) **Temporal-context restriction** is the creation of relevant contexts (e.g., effect of a drug) for interpretation of data, crucial for limiting the scope of the inference; 2) **Vertical temporal inference** is inference from values of contemporaneous input data or abstractions into values of higher-level concepts (e.g., classification of the results of several blood tests conducted during the same day into bone-marrow toxicity Grade II); 3) **Horizontal temporal inference** is inference from similar-type propositions that hold over different time intervals (e.g., joining different-value abstractions of the same clinical parameter that hold over two meeting time intervals, and computing the value of the new abstraction); 4) **Temporal interpolation** is the bridging of gaps between similar-type but temporally disjoint point- or interval-based propositions to create longer intervals (e.g., joining two disjoint episodes of anemia, occurring on different days, into a longer episode); 5) **Temporal-pattern matching** is the creation of intervals by matching of patterns over disjoint intervals, over which hold various propositions (e.g., onset of quiescent-onset chronic graft-versus-host disease).

The RÉSUMÉ software system forms temporal abstractions given time-stamped patient data and a domain-specific knowledge base.[5] RÉSUMÉ has been evaluated with encouraging results in a variety of

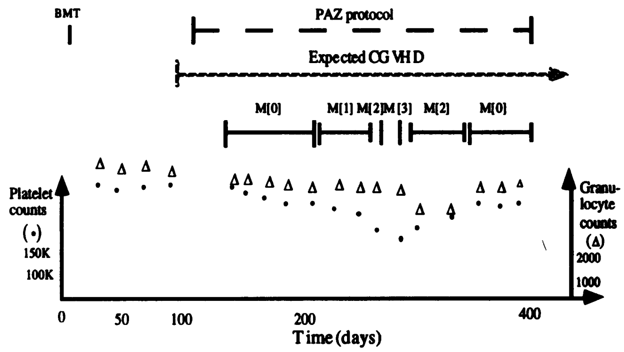

Figure 1: Typical inputs to and outputs of the temporal-abstraction task. The figure presents examples of abstractions of platelet and granulocyte values during administration of the PAZ clinical protocol for treating patients who have chronic graft-versus-host disease (CGVHD). The time line starts with a bone-marrow-transplantation (BMT) event. ⊢ ⊣ = event; • = platelet counts; Δ = granulocyte counts; ⊢→ = open context interval; ⊢⊣ = closed abstraction interval; M[n] = myelotoxicity (bone-marrow–toxicity) grade *n*.

clinical domains, such as protocol-based care, monitoring of children's growth,[6] and management of insulin-dependent diabetes.[7]

DOMAIN-SPECIFIC KNOWLEDGE

The domain-specific knowledge required by the temporal-abstraction mechanisms is represented as a **parameter-properties ontology**: a representation of the raw and abstract **parameters** (e.g., blood glucose level) in that domain, their temporal properties (e.g., persistence over time), and the relations among them.[8] The parameter-properties ontology is used by all the temporal-abstraction mechanisms. The context-forming mechanism also refers to an ontology of external events and an ontology of interpretation contexts. An **event** is any external occurrence or intervention (e.g., insulin administration) that affects a clinically relevant parameter. **Interpretation contexts** are induced by events or parameter abstractions, and alter the interpretation of concurrent parameter values (e.g., the context of chemotherapy modifies the interpretation of hematological values). To be useful for a particular clinical domain, the temporal-abstraction mechanisms must take as input domain-specific knowledge. This domain-specific knowledge is the *only* interface between the KBTA method and the system developer. Thus, constructing a temporal-abstraction system particular to a new domain requires only creating or editing a predefined set of knowledge categories.

We distinguish among four **knowledge types** (Figure 2) used by the temporal abstraction mechanisms:

1. **Structural knowledge** consists of the IS-A and PART-OF relations that link concepts in a domain. For example, the parameter WHITE_BLOOD_CELL_COUNT has an IS-A relation to the more general class HEMATOLOGICAL_PARAMETER.

2. **Classification knowledge** allows parameters to be assigned values, such as the classification of blood glucose levels into HYPOGLYCEMIC, LOW, NORMAL, HIGH.

3. **Temporal–semantic knowledge** allows inferences to made about temporal intervals. For example, two abstraction intervals that share the *concatenable* property can be joined into a single superinterval.

4. **Temporal–dynamic knowledge** includes properties such as the persistence of the value of a parameter over time.

THE KNOWLEDGE-ACQUISITION TOOL

In previous trials of RÉSUMÉ, a knowledge engineer captured the necessary domain-specific knowledge in a series of interviews with a domain expert, and then coded it with a text editor in a form readable by the system. Knowledge acquisition typically necessitated four to six 2-hour interviews, followed by 1 to 2 weeks of encoding. Subsequent modifications to the knowledge base had to be performed with a text editor by a person familiar with the RÉSUMÉ knowledge structures.

We took advantage of the PROTÉGÉ-II framework in designing the KA tool. **PROTÉGÉ-II**, a development environment for knowledge-based expert systems, supports libraries of reusable problem-solving methods.[7,9] RÉSUMÉ is one such method. PROTÉGÉ-II provides tools for constructing formal ontologies of task-specific knowledge, for generating and tailoring graphical KA tools from these ontologies, and for using the resulting KA tool to acquire and maintain domain-specific knowledge.

Knowledge acquisition is largely a problem of modeling. Thus, the first step in using PROTÉGÉ-II to construct a KA tool was developing a KA ontology (Figure 3): an explicit conceptualization of the

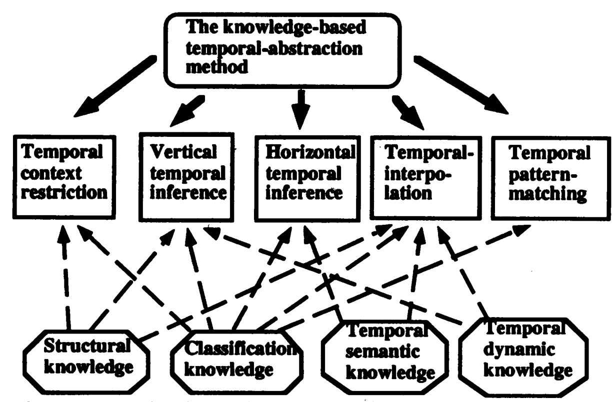

Figure 2: The knowledge-based temporal-abstraction method. ▭ = task; ⬭ = method; ⬡ = knowledge type; ➔ = DECOMPOSED-INTO relation; - -➔ = USED-BY relation.

knowledge to be acquired from a domain expert. This formal declaration of knowledge roles is necessary to provide a structure to the user interface of the eventual KA tool, and also to make clear the necessary mappings between the output of the tool and requirements of RÉSUMÉ. Typically, a developer begins designing a knowledge-based system with PROTÉGÉ-II by constructing a method-independent, domain-specific ontology, then selecting a problem-solving method best suited to the task at hand.[4] She then expands or modifies the domain ontology to encompass the knowledge requirements of the selected method. However, because RÉSUMÉ has a large set of knowledge requirements, and because the knowledge specific to the temporal-abstraction task is not normally included in a method-independent domain ontology, we chose instead to use RÉSUMÉ's input–output requirements as a starting point for the KA tool. The resulting method-specific KA ontology, modeled on RÉSUMÉ's internal, hard-coded ontology, can be tailored for any medical domain.

We constructed the KA ontology using the PROTÉGÉ-II graphical editor for creating, browsing, and editing class hierarchies. This formal specification of the items and relations to be acquired was designed to match as closely as possible RÉSUMÉ's knowledge structures. However, the KA ontology has the constraint that, because it provides the basis of a KA tool, its surface structure has to be easily comprehensible by an expert clinician who is not acquainted with the details of RÉSUMÉ, and it must guide the user of the resulting tool through a clear and consistent dialogue. Furthermore, some of the complex knowledge structures assumed by RÉSUMÉ cannot be acquired directly because of constraints imposed by the PROTÉGÉ-II tools. For example, RÉSUMÉ internally makes use of three-dimensional tables that are difficult to represent graphically. Thus, development of the KA ontology for RÉSUMÉ necessitated many important design decisions. Because the KA tool is an intermediary layer between a domain expert and an expert system, it must meet the needs of both the user and the computational method. We had to model complicated relationships as simply as possible, and break complex knowledge structures into smaller, more readily acquired pieces, to create a suitable user interface. Likewise, we had to represent the full range of RÉSUMÉ's knowledge requirements in the KA ontology to fulfill the goal of instantiating RÉSUMÉ in multiple clinical domains. We continue to refine the KA ontology both to model RÉSUMÉ's knowledge structures more accurately, and to take advantage of updated features of PROTÉGÉ-II.

Mappings

The KA tools generated by PROTÉGÉ-II acquire and store knowledge as collections of user-defined instances of the classes specified in the KA ontology. If a problem-solving method is to be able to use the resulting knowledge base, the terms and relations in the method must be mapped to the corresponding terms and relations in the KA ontology,[3] and any necessary translation must be performed.

Because of the design issues discussed earlier, the temporal-abstraction KA ontology differs from RÉSUMÉ's knowledge structures in small but important ways (Figure 3). The class structure is arranged to present a more simple and uniform view to the user of the KA tool. Large knowledge structures are broken into several pieces to enhance the user interface. To perform the necessary mappings, we constructed a filter that preprocesses the output of the KA tool into a form readable by RÉSUMÉ. The ontologies of both RÉSUMÉ and the KA tool might change in the future, so we designed the filter to be reusable and easily modified.

The filter itself consists of a general algorithm onto which are attached small modular pieces of code that perform the actual transformation. The filter reads in

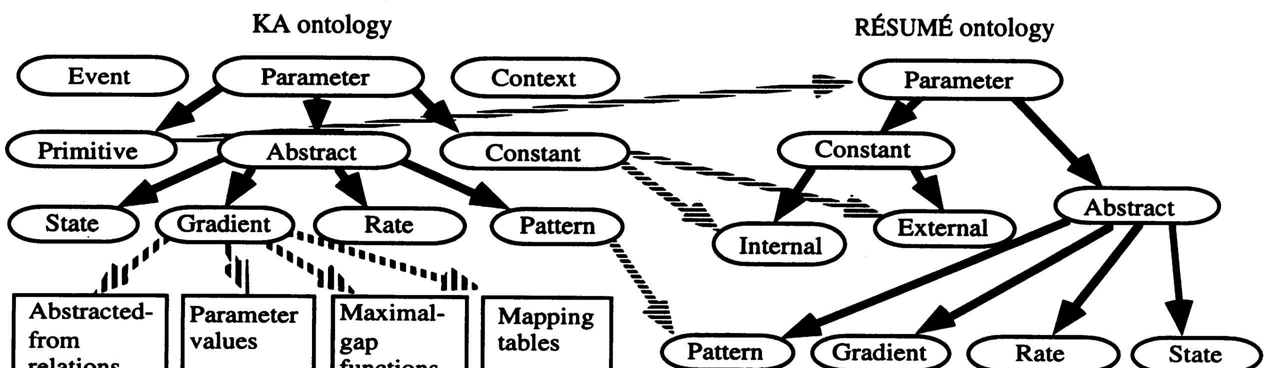

Figure 3: Part of the ontologies of the KA tool and RÉSUMÉ, and some of the mappings between them. = class; = property; = IS-A relation; = PROPERTY-OF relation; = MAPPED-TO relation. The KA tool captures domain-specific knowledge by creating instances of the classes in the KA ontology. For example, in the diabetes domain, GLUCOSE_STATE is an instance of a *state parameter* and has an ABSTRACTED-FROM relation to the *primitive parameter* GLUCOSE. The tool then maps this knowledge to the knowledge structures in RÉSUMÉ for use in temporal abstraction. For clarity, only a few of the mappings are shown.

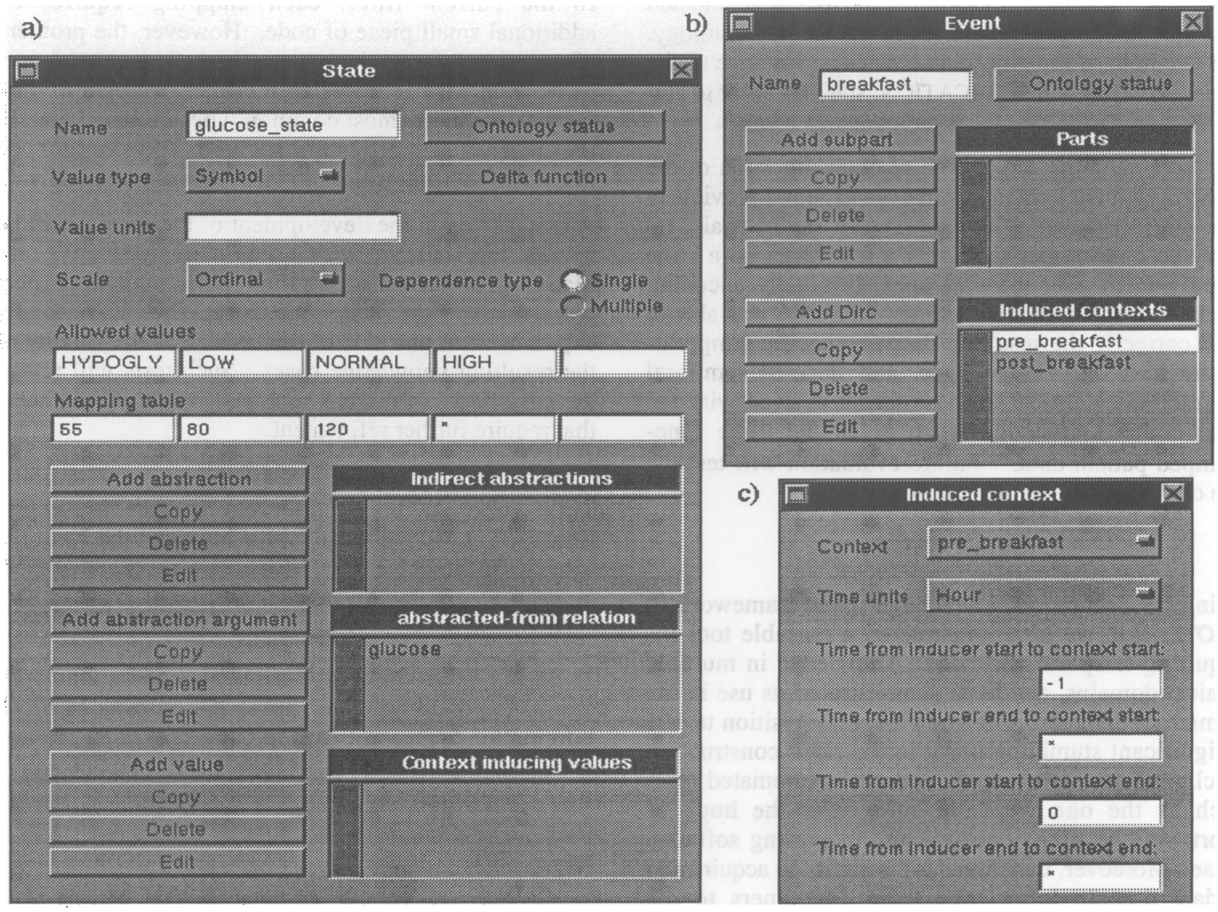

Figure 4: Graphical forms from the KA tool. (a) An example of a state parameter. As shown, GLUCOSE_STATE has an ABSTRACTED-FROM relation to GLUCOSE. The mapping table contains the classification knowledge needed to abstract the correct value of the state parameter (e.g., a GLUCOSE value of 55 to 80 results in a GLUCOSE_STATE abstraction with value LOW). (b) A form for acquiring events. The user can enter new instances in the list browsers, or review previously entered knowledge. For example, if the user selects the induced context PRE_BREAKFAST and clicks on the edit button , then (c) the corresponding form appears. The user is then free to edit that form.

the knowledge acquired with the KA tool, and then iteratively calls the pieces of code. Each of these small modular programs represents an individual mapping from a class in RÉSUMÉ's ontology to an instance or set of instances acquired by the KA tool. The result is an output file of knowledge structures compatible with the RÉSUMÉ ontology.

Any change in the RÉSUMÉ or KA ontologies will necessitate a change in only the corresponding mapping code, rather than in the filter itself. we have deliberately kept the mappings as straightforward as possible. Sometimes, no transformation is required, or, in the simplest case, an integer might be converted to a floating-point number. Other structures, however, require more processing. For example, as mentioned earlier, RÉSUMÉ internally makes use of three-dimensional tables that the KA tool must represent graphically with vectors and two-dimensional tables. A more complex function is needed to translate between the two representations. The advantages of using lightweight, independent pieces of code are that the source of an error can be quickly located and fixed; individual mappings can be easily removed, augmented, or replaced; and the filter can rapidly adjust to changes in either of the two ontologies. Moreover, the filter's algorithm is completely independent of any given domain or method, and therefore is suitable for mapping between any two ontologies.

RESULTS

Given the KA ontology, PROTÉGÉ-II automatically generates a graphical KA tool[4] (Figure 4). Initially, PROTÉGÉ-II creates a generic layout, which the knowledge engineer can tailor by repositioning the elements and editing the text labels. Standard graphical metaphors — such as radio buttons, text fields, pop-up menus, and list browsers — allow the user to create, to browse, and to edit a knowledge base.

The KA tool includes forms for acquiring primitive, state, gradient, rate, and pattern parameters, as well as contexts and events. Each form corresponds to an instance of a class in the method ontology, and the input data correspond to slot values. These forms pop up automatically when the user indicates that she wants to enter a new instance. In addition, the user is able to add new slots to the generic classes in the method ontology. For example, if the physician creates an instance of the class EVENT called MEDICATION, she may want to add an attribute DOSE that takes an integer as a value.

We have used the KA tool to re-enter a portion of the diabetes domain knowledge that we acquired previously through interviews with experts. In initial trials, reentry by a developer took only a few hours — a large improvement over the time spent originally encoding the knowledge with a text editor. RÉSUMÉ was able to read correctly the resulting domain-specific temporal-abstraction knowledge base, and derived temporal abstractions identical to those that it derived with the hand-coded knowledge when given the same time-stamped patient data.[7] Future evaluation will test the use of the tool by domain experts.

DISCUSSION

Using the declarative, knowledge-based framework of PROTÉGÉ-II, we have constructed a reusable tool for acquiring temporal-abstraction knowledge in multiple clinical domains, and have demonstrated its use in the domain of diabetes. The knowledge-acquisition task is a significant stumbling block in the rapid construction of clinical decision-support systems. Automated tools such as the one we have built offer the hope of shortening development time and increasing software reuse. Moreover, they transfer the ability to acquire and update domain knowledge from developers to the expert physicians who actually use the systems. These features give problem-solving methods the flexibility to be applied quickly to a wide variety of medical domains, and to evolve as domain knowledge changes.

The ease of use of a knowledge-based software component depends critically on the effective design of that component's ontology. In general, it is not enough that the knowledge structures be declarative. They should also be clear, intuitively structured, and broken into manageable pieces for knowledge acquisition. In addition to providing a knowledge framework for an underlying piece of software, an ontology acts as an interface — with system developers, with domain experts, and with other programs. Thus, the optimal design for an ontology is not necessarily the one that is most compact or efficient for a single given use. For example, in the case of PROTÉGÉ-II, it is important that the KA ontology be as similar as possible to the problem-solving method's ontology, even if the KA tool is domain specific, to ensure that mappings will be simple and robust.

Because the KA tool interface is constrained by the windowing and development environments, knowledge engineers should design the problem-solving method with knowledge acquisition in mind. They should minimize implicit semantic constraints (i.e., dependencies) between items in the method ontology, and keep structures as simple as possible.

In the current filter, each mapping requires an additional small piece of code. However, the problem of mapping between knowledge bases is a general one; as we gain experience using the filter, we plan to look for patterns in the most common types of mappings. In the future, a more sophisticated filter might be instructed to perform certain mappings automatically.

A crucial step in the development of the system will be to test that system with domain experts. We are planning a series of studies with several expert physicians that will measure the KA tool's expressiveness and ease of use, as well as the validity of the resulting knowledge bases. Such tests will reveal the advantages of the KA tool, and will highlight areas that require further refinement.

Acknowledgments

This work has been supported in part by grants LM05305, LM05708, and LM06245 from the National Library of Medicine. Dr. Musen is a recipient of NSF Young Investigator Award IRI-9257578.

References

1. Shahar Y. A knowledge-based method for temporal abstraction of clinical data. Ph.D. dissertation; Medical Information Sciences, Stanford University, CA. 1994

2. Guarino N, Giaretta P. Ontologies and knowledge bases: Towards a terminological clarification. *Towards Very Large Knowledge Bases*, N. J. I. Mars (ed.), IOS Press, 1995.

3. Gennari JH, Tu SW, Rothenfluh TE, Musen MA. Mapping domains to methods in support of reuse. *International Journal of Human-Computer Studies*, 1994; 41:399–424.

4. Musen MA, Gennari JH, Eriksson H, Tu SW, Puerta AR. PROTÉGÉ-II: Computer support for development of intelligent systems from libraries of components. *Proceedings of MEDINFO 95, The Eighth World Congress on Medical Informatics,* 1995; 766–70.

5. Shahar Y, Musen MA. RÉSUMÉ: A temporal-abstraction system for patient monitoring. *Computers and Biomedical Research,* 1993; 26:255–73.

6. Kuilboer MM, Shahar Y, Wilson DM, Musen MA. Knowledge reuse: Temporal-abstraction mechanisms for the assessment of children's growth. *Proceedings, Seventeenth Annual Symposium on Computer Applications in Medical Care*, 1993; 449–53.

7. Shahar Y, Musen MA. Knowledge-based temporal abstraction in clinical domains. *Artificial Intelligence in Medicine,* 1996; 8:267–98.

8. Shahar Y, Tu SW, Musen MA. Knowledge acquisition for temporal-abstraction mechanisms. *Knowledge Acquisition,* 1992; 4:217–36.

9. Tu SW, Eriksson H, Gennari JH, Shahar Y, Musen MA. Ontology-based configuration of problem-solving methods and generation of knowledge-acquisition tools: Application of PROTÉGÉ-II to protocol-based decision support. *Artificial Intelligence in Medicine,* 1995; 7:257–89.